[NAV TechDays 2017 - CAL, Coding for Performance](https://www.youtube.com/watch?v=C4hMN0-GYHQ)

**15:20 – Minimize server roundtrips**

Excessive roundtrips don’t have a footprint, memory or processor looks good but a query can run for ever.

Replace indented loops with queries. A query gives just one roundtrip.

Use SETAUTOCALCFIELDS outside of the loop.

Use CALCSUMS, MODIFYALL, DELETEALL instead of loops.

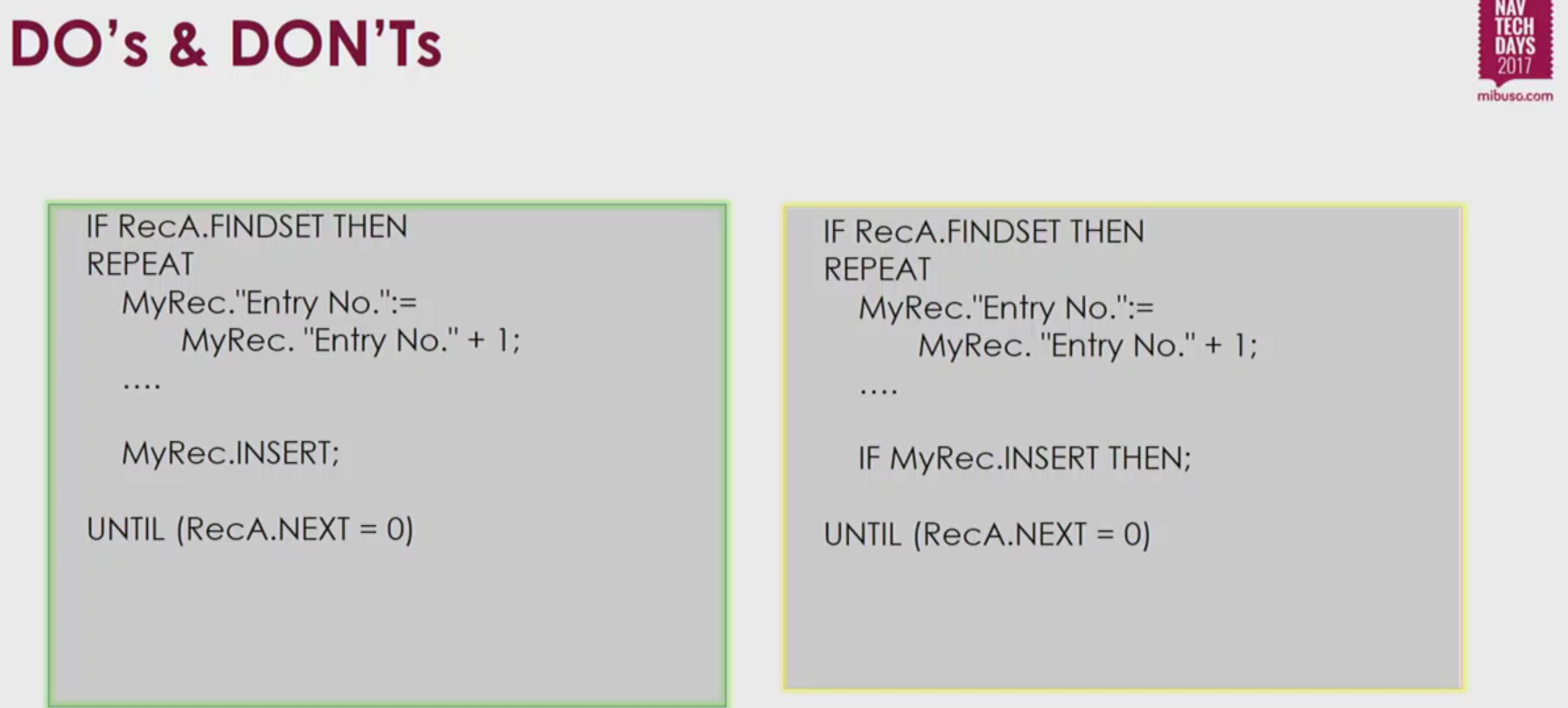
**19:00 – Data retrieval methods**

FIND: use it when you are not sure if you are going to read all the records. Always two loops (roundtrips).

FINDSET: when you are sure that you’ll read all the records. Always one loop (roundtrip). It retrieves all the records in that one loop so if you won’t read them you put unnecessary memory pressure to SQL. Also, use this when modifying records.

FINDFIRST,FINDLAST,ISEMPTY: one roundtrip.

**24:45**

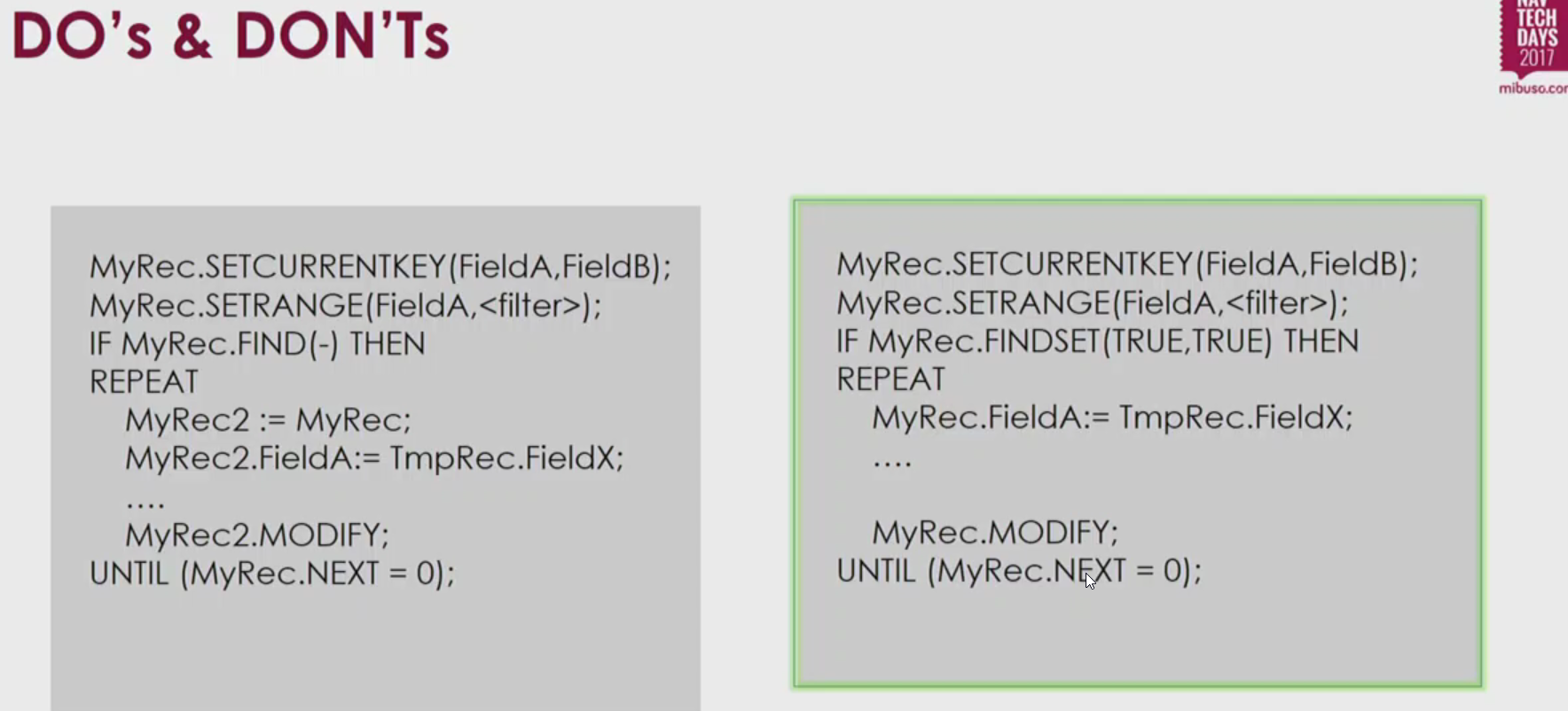


Left-hand side makes a buffered insert = 1 roundtrip. NAV will collect all inserts and send to SQL after the completion of loop.

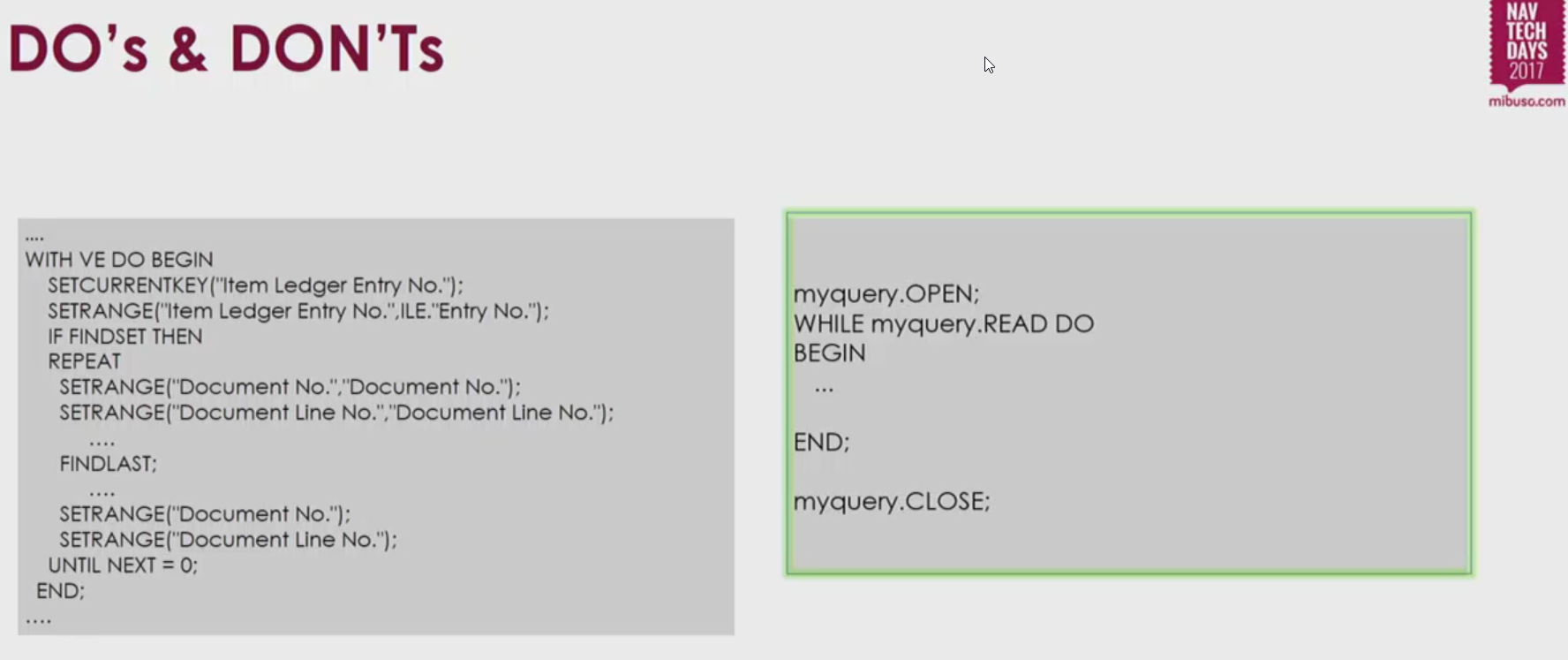
Right-hand side breaks buffered insert = multiple roundtrips

Detail: when you have very wide records the benefit of buffered inserts levels out. In such cases it might be beneficial to break buffered inserts by IF rec.INSERT THEN.

**28:30**



**30:00**

****

Left-hand side is actually a “Select Distinct” and it’s performance killer.

Check [SELECT DISTINCT pattern](https://community.dynamics.com/nav/w/designpatterns/152.select-distinct-with-queries)

**35:00**

Subscribers: subscribe always to the most granular events.

MARK: a roundtrip per mark -> avoid

GUIDS: have benefits but also performance impact.

**47:00** Performance in SQL server has to do with retrieval.

**1:14:55** Most important performance indicator is the number of reads

**1:16:00** Vast majority of expensive queries can be fixed by index optimization.



“Navision indexes shucks. SQL indexes rocks”

If you add indexes at a SQL level use the following script to review and fine tune periodically.

